

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of:)	
)	
STEVENS)	
)	
Serial No.: 09/766,335)	Group Art Unit: 2178
)	
Filed: January 19, 2001)	Examiner: C. Huynh
)	
For: CONVERSION SYSTEM FOR)	Board of Patent Appeals and
TRANSLATING STRUCTURED)	Interferences
DOCUMENTS INTO MULTIPLE)	
TARGET FORMATS)	
)	
Confirmation No.: 7723)	

Mail Stop: Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF UNDER 37 C.F.R. § 41.37

In support of the Notice of Appeal filed on August 20, 2008, and pursuant to 37 C.F.R. § 41.37, Appellant presents this Appeal Brief in the above-captioned application.

This is an appeal to the Board of Patent Appeals and Interferences from the Examiner's final rejection of claims 1, 4-20, and 22-38 in the Final Office Action dated May 22, 2008. The appealed claims are set forth in the attached Claims Appendix.

1. Real Party in Interest

This application is assigned to Wind River Systems, Inc., the real party in interest.

2. Related Appeals and Interferences

There are no other appeals or interferences that would directly affect, be directly affected, or have a bearing on the instant appeal.

3. Status of the Claims

Claims 1, 4-20, and 22-38 have been rejected in the Final Office Action. Claims 2, 3, and 21 were canceled in a previous amendment. The final rejection of claims 1, 4-20, and 22-38 is being appealed.

4. Status of Amendments

All amendments submitted by Appellant have been entered.

5. Summary of Claimed Subject Matter

The present invention, as recited in independent claim 1, relates to a translator 100, embodied in a computer readable storage medium, for translating a source file 110 in a source format to a target file 112 in a target format. (Specification at [0033]; Figure 1). The translator 100 comprises a feature identifier 114 to determine a feature set of the source file 110 (Specification at [0034]; Figure 1); a buffer to assemble the feature set (Specification at [0034]; Figure 1); a feature writer 118 to write the feature set into the target file 112 in the target format (Specification at [0034]; Figure 1); and an output module to output the target file.

The present invention, as recited in independent claim 20, relates to a method of translating a file from a source format 110 to a target format 112. (Specification at [0033]; Figure 1). The method comprises identifying a feature set of a source file 110 (Specification at [0034]; Figure 1); assembling the feature set in a buffer (Specification at [0034]; Figure 1); and writing the feature set into a target file 112 in the target format 112. (Specification at [0034]; Figure 1).

The present invention, as recited in independent claim 34, relates to a method of configuring a system to translate a source file 110 in a source format 110 to a target file 112 in a

target format 112. (Specification at [0033]; Figure 1). The method comprises providing a feature identifier 114 to determine a feature set of the source file 110 (Specification at [0034]; Figure 1); providing a buffer to assemble the feature set (Specification at [0034]; Figure 1); and providing a feature writer 118 to write the feature set into the target file 112 in the target format 112. (Specification at [0034]; Figure 1).

The present invention, as recited in independent claim 35, relates to a system for translating a source file 110 in a source format 110 to a target file 112 in a target format 112. (Specification at [0033]; Figure 1). The system comprises a feature identifier 114 to determine a feature set of the source file 110 (Specification at [0034]; Figure 1); a buffer to assemble the feature set (Specification at [0034]; Figure 1); a feature writer 118 to write the feature set into the target file 112 in the target format 112 (Specification at [0034]; Figure 1); and an output module to output the target file.

The present invention, as recited in independent claim 36, relates to an article of manufacture for translating a source file 110 in a source format 110 to a target file 112 in a target format 112. (Specification at [0033]; Figure 1). The article of manufacture comprises a computer usable medium having a computer readable program code embodied therein, the computer usable medium having: computer readable program code for identifying a feature set of the source file 110 (Specification at [0034]; Figure 1); computer readable program code for assembling the feature set in a buffer (Specification at [0034]; Figure 1); and computer readable program code for writing the feature set into the target file 112 in the target format 112. (Specification at [0034]; Figure 1).

The present invention, as recited in independent claim 37, relates to a computer readable program code for translating a source file 110 in a source format to 110 a target file 112 in a target format 112. (Specification at [0033]; Figure 1). The computer readable program code comprises computer readable program code for identifying a feature set of the source file 110 (Specification at [0034]; Figure 1); computer readable program code for assembling the feature set in a buffer (Specification at [0034]; Figure 1); and computer readable program code for writing the feature set into the target file 112 in the target format 112. (Specification at [0034]; Figure 1).

The present invention, as recited in independent claim 38, relates to a translator 100, embodied in a computer readable storage medium, for translating a source file 110 in an MIF format to a target file 112 in an HTML format. (Specification at [0033]; Figure 1). The translator comprises a feature identifier 114 having a front-end lookup table to map MIF code fragments of the source file to a list of features to determine a feature set of the source file 110 (Specification at [0034], [0038]; Figure 1); a buffer to store and assemble the feature set (Specification at [0034]; Figure 1); a feature writer 118 having a back-end lookup table to map the feature set to HTML code fragments, to write the feature set into the target file in the HTML format (Specification at [0040]; Figure 1); and an output module to output the target file.

6. Ground of Rejection to be Reviewed on Appeal

I. Whether Claims 1, 4-20 and 22-38 are anticipated under 35 U.S.C. § 102(e) by U.S. Patent Publication No. 2002/0052893 to Grobler et al. (hereinafter "Grobler").

7. Argument

Claim 20 recites, "(a) identifying a feature set of a source file; (b) assembling the feature set in a buffer; and (c) writing the feature set into a target file in the target format." The Examiner asserts that the above recitation of claim 20 is taught in Grobler in figures 8 and 9 and paragraphs [0067]-[0069] and [0071]. (See 05/22/08 Office Action, p. 2). Applicant respectfully submits that neither the above referenced portions, nor any other portions, of Grobler teach or suggest the above recitation of claim 20.

The Examiner equates the "assembling the feature set in a buffer," in claim 20, to the temporary storage of the source data in Grobler. However, this is incorrect. Grobler states:

[a]fter the source and target of the import process have been selected in operation 810, the source data selected in operation 810 is handed over to and temporarily stored in temporarily store source data operation 820. Next analyze source data structure operation 830 analyzes the table structure of the temporarily stored source data. Operation 830 parses the source data for tags, which indicate the table structure of the source data.

(See Grobler par. [0063]-[0064]). Grobler takes the source data, and then stores it, in its entirety, in a temporary storage space. After the source data has been moved to the temporary storage space the source data is analyzed for tags. In contrast, claim 20 specifically recites, “(a) identifying a feature set of a source file; (b) assembling the feature set in a buffer.” The identifying of the feature set in the source file is done prior to being stored in a buffer. Once the feature set has been identified, the feature set is then moved into a buffer. In Grobler, however, the source data is moved into temporary storage and then analyzed for tags. Therefore, Applicant respectfully submits that Grobler does not teach or suggest the above recitation of claim 20. Thus, Applicant respectfully submits that claim 20 is patentable over Grobler. Because claims 22-33 depend from, and therefore include all the limitations of claim 20, it is respectfully submitted that these claims are also allowable for at least the same reasons given above with respect to claim 20.

In the Advisory Action, the Examiner, after referring to the identifying and assembling steps, states that “it just means that after the identifying step, the feature set, is assembled in the buffer. Claim 20 does not necessarily mean that the identifying of the feature set in the source file is done prior to being stored in a buffer.” Advisory Action at page 2. Appellant does not know what this means. The Examiner appears to contradict himself because if the feature set is assembled in the buffer “after the identifying step,” then it cannot be denied logically that the identifying step is done before the assembling step. Moreover, the Examiner states that in “Gobbler [sic], analyzing the tags for the table structure in the source data and selecting only tags suitable for a target table to import to the target file...are considered equivalent to assembling the feature set in a buffer since said steps are carried for combining table tags in the temporary memory.” *Id.* Appellant respectfully points out that the rejection on appeal is one of anticipation, meaning that every feature of the claims must be identically taught by the prior art. Here, in this quote, the Examiner is basing the rejection on a different standard, equivalency, which is not the same as identity. Therefore, to this extent, the Examiner has evaluated the relevancy of Grobler under Section 102 using the wrong standard of comparison. Accordingly, for these reasons, reversal of the rejection is requested.

Independent claim 34 recites, “(a) providing a feature identifier to determine a feature set of the source file; (b) providing a buffer to assemble the feature set.” Applicant respectfully submits that this claim is also allowable for at least the same reasons stated above with respect to claim 20.

Independent claim 35 recites, “a feature identifier to determine a feature set of the source file; a buffer to assemble the feature set.” Applicant respectfully submits that this claim is also allowable for at least the same reasons stated above with respect to claim 20.

Independent claim 36 recites, “computer readable program code for identifying a feature set of the source file; computer readable program code for assembling the feature set in a buffer.” Applicant respectfully submits that this claim is also allowable for at least the same reasons stated above with respect to claim 20.

Independent claim 37 recites, “computer readable program code for identifying a feature set of the source file; computer readable program code for assembling the feature set in a buffer.” Applicant respectfully submits that this claim is also allowable for at least the same reasons stated above with respect to claim 20.

Independent claim 1 recites, “a feature identifier to determine a feature set of the source file; a buffer to assemble the feature set.” Applicant respectfully submits that this claim is also allowable for at least the same reasons stated above with respect to claim 20. Because claims 4-19 depend from, and therefore include all the limitations of claim 1, it is respectfully submitted that these claims are also allowable for at least the same reasons given above with respect to claim 1.

Independent claim 38 recites, “a feature identifier having a front-end lookup table to map MIF code fragments of the source file to a list of features to determine a feature set of the source file; a buffer to store and assemble the feature set; computer readable program code for assembling the feature set in a buffer.” Applicant respectfully submits that this claim is also allowable for at least the same reasons stated above with respect to claim 20.

8. Conclusion

For the reasons set forth above, Appellant respectfully requests that the Board reverse the rejection of the claims by the Examiner under 35 U.S.C. § 102(e) and indicate that claims 1, 4-20, and 22-38 are allowable.

Respectfully submitted,

Date: October 20, 2008

By: L82 mag
Dervis Magistre (Reg. No. 41,172)

Fay Kaplun & Marcin, LLP
150 Broadway, Suite 702
New York, NY 10038
Tel.: (212) 619-6000
Fax: (212) 619-0276

CLAIMS APPENDIX

1. (Previously presented) A translator, embodied in a computer readable storage medium, for translating a source file in a source format to a target file in a target format, the translator comprising:
 - a feature identifier to determine a feature set of the source file;
 - a buffer to assemble the feature set;
 - a feature writer to write the feature set into the target file in the target format; and
 - an output module to output the target file.
4. (Previously presented) The translator of claim 1, wherein features of the feature set are selected from the group consisting of paragraph style, straddled cells in a table, cross-referencing, pen styles in a drawing, other document formatting, document header specifications, document footer specifications, discontinuity indicators, order indicators, location indicators, beginning indicators, ending indicators, data types, data translation pairs, document macros, user-created features, implied feature endings and combinations thereof.
5. (Original) The translator of claim 1, wherein the feature identifier comprises a front-end converter to map code fragments of the source file to a list of features.
6. (Original) The translator of claim 5, wherein the feature identifier comprises a front-end lookup table.
7. (Original) The translator of claim 6, wherein the front-end lookup table is user modifiable.
8. (Original) The translator of claim 1, wherein the feature writer comprises a back-end converter to map the feature set to code fragments of the target file format.
9. (Original) The translator of claim 8, wherein the back-end converter comprises a back-end lookup table.

10. (Original) The translator of claim 5, comprising a plurality of feature writers to write the feature set into a plurality of target files having a plurality of target formats.
11. (Original) The translator of claim 1, comprising a plurality of feature identifiers to determine a feature set of a plurality of source files having a plurality of source formats.
12. (Original) The translator of claim 5, wherein the front-end converter comprises a lexical analyzer to identify tokens disposed within the source file, and a feature collector to associate the tokens with the feature set.
13. (Original) The translator of claim 1, further comprising a user interface.
14. (Original) The translator of claim 13, wherein the user interface comprises a GUI.
15. (Original) The translator of claim 1, further comprising a source format adapter module to interface with a source file generator.
16. (Original) The translator of claim 15, wherein the source format adapter module enables the source file generator to initiate translation by the translator.
17. (Original) The translator of claim 1, further comprising a target file adapter module to perform secondary translation.
18. (Original) The translator of claim 17, wherein the target file adapter module translates the target file into another target format.
19. (Original) The translator of claim 1, wherein the source and target formats are selected from the group consisting of MIF, RTF, WordPerfect, VENTURA, Microsoft Word, Interleaf, HTML, SGML, SML, C, C++, Visual Basic, Pascal, Java, MFC, PowerPlant, Swing, SVG, HPJ, Flash, WMF, VRML, RenderMan, 3DMF, and combinations thereof.

20. (Previously presented) A method of translating a file from a source format to a target format, the method comprising:
- (a) identifying a feature set of a source file;
 - (b) assembling the feature set in a buffer; and
 - (c) writing the feature set into a target file in the target format.
22. (Previously presented) The method of claim 20, wherein features of the feature set are selected from the group consisting of paragraph style, straddled cells in a table, cross-referencing, pen styles in a drawing, other document formatting, document header specifications, document footer specifications, discontinuity indicators, order indicators, location indicators, beginning indicators, ending indicators, data types, data translation pairs, document macros, user-created features, implied feature endings and combinations thereof.
23. (Original) The method of claim 20, wherein the identifying step (a) comprises mapping code fragments of the source file to a feature list.
24. (Original) The method of claim 23, wherein the identifying step (a) comprises looking up the code fragments in a front-end lookup table.
25. (Original) The method of claim 24, further comprising permitting the front-end lookup table to be user modifiable.
26. (Original) The method of claim 20, wherein the writing step (b) comprises mapping the feature set to code fragments of the target file format.
27. (Original) The method of claim 26, wherein the writing step (b) comprises looking up the feature set in a back-end lookup table.
28. (Original) The method of claim 20, wherein the writing step (b) comprises writing the feature set into a plurality of target files having a plurality of target formats.

29. (Original) The method of claim 20, wherein the identifying step (a) comprises identifying a feature set of a plurality of source files having a plurality of source formats.
30. (Original) The method of claim 20, wherein the identifying step (a) comprises identifying tokens disposed within the source file, and associating the tokens with the feature list.
31. (Original) The method of claim 20, further comprising using a source file generator to initiate translation by the translator.
32. (Original) The method of claim 20, further comprising using a target file adapter module to perform secondary translation.
33. (Original) The method of claim 32, wherein the target file adapter module translates the target file into another target format.
34. (Previously presented) A method of configuring a system to translate a source file in a source format to a target file in a target format, the method comprising:
- (a) providing a feature identifier to determine a feature set of the source file;
 - (b) providing a buffer to assemble the feature set; and
 - (c) providing a feature writer to write the feature set into the target file in the target format.
35. (Previously presented) A system for translating a source file in a source format to a target file in a target format, the system comprising:
- a feature identifier to determine a feature set of the source file;
 - a buffer to assemble the feature set;
 - a feature writer to write the feature set into the target file in the target format; and
 - an output module to output the target file.

36. (Previously presented) An article of manufacture for translating a source file in a source format to a target file in a target format, the article of manufacture comprising:

a computer usable medium having a computer readable program code embodied therein,
the computer usable medium having:

computer readable program code for identifying a feature set of the source file;
computer readable program code for assembling the feature set in a buffer; and
computer readable program code for writing the feature set into the target file in

the target format.

37. (Previously presented) Computer readable program code for translating a source file in a source format to a target file in a target format, the computer readable program code comprising:

computer readable program code for identifying a feature set of the source file;
computer readable program code for assembling the feature set in a buffer; and
computer readable program code for writing the feature set into the target file in the

target format.

38. (Previously presented) A translator, embodied in a computer readable storage medium, for translating a source file in an MIF format to a target file in an HTML format, the translator comprising:

a feature identifier having a front-end lookup table to map MIF code fragments of the source file to a list of features to determine a feature set of the source file;

a buffer to store and assemble the feature set;

a feature writer having a back-end lookup table to map the feature set to HTML code fragments, to write the feature set into the target file in the HTML format; and
an output module to output the target file.

EVIDENCE APPENDIX

No evidence has been submitted herewith or is relied upon in the present appeal.

RELATED PROCEEDINGS APPENDIX

There are no related proceedings or decisions which relate to the present appeal.